Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (original) A method for combining lookup tables (LUTs) in a design for a programmable integrated circuit, the method comprising:

selecting first and second LUTs from the design;

determining whether both of the LUTs implement a same function; and

if the first and the second LUTs implement the same function, combining masks of the LUTs into a shared LUT mask in the design.

2. (original) The method of claim 1 further comprising:

before determining whether the LUTs implement the same function, identifying common input signals between the first and the second LUTs; and

rearranging an order of input signals of one of the LUTs so that each of the common input signals is applied to a corresponding input terminal in both of the LUTs to provide a first rearranged order of the input signals,

wherein the method determines whether the LUTs implement the same function based on the first rearranged order of the input signals.

3. (original) The method of claim 2 further comprising:

if the LUTs do not implement the same function, rearranging an order of at least two uncommon input signals of one of the LUTs with respect to input terminals of that LUT to provide a second rearranged order; and

determining if both LUTs implement the same function based on the second rearranged order of the input signals.

4. (original) The method of claim 1 further comprising:

if the LUTs do not implement the same function, rearranging an order of at least two input signals of the first LUT with respect to input terminals of the first LUT to provide a first rearranged order; and

determining if both LUTs implement the same function based on the first rearranged order of the first LUT input signals.

5. (original) The method of claim 4 further comprising:

if the LUTs implement the same function with the first rearranged order, combining the masks of the LUTs into a shared LUT mask in the design; and

if the LUTs do not implement the same function with the first rearranged order, rearranging the order of at least two input signals of the first LUT with respect to the input terminals of the first LUT to provide a second rearranged order.

6. (original) The method of claim 5 further comprising:

determining if both LUTs implement the same function based on the second rearranged order of the first LUT input signals;

if the LUTs implement the same function with the second rearranged order, combining masks of the LUTs into a shared LUT mask in the design; and

if the LUTs do not implement the same function with the second rearranged order, rearranging the order of at least two input signals of the first LUT with respect to the input terminals of the first LUT to provide a third rearranged order.

7. (currently amended) The method of claim 1 further comprising:

before determining whether the LUTs implement the same function, determining if the LUTs have at least N common input signals; and

if the LUTs do not have at least N common input signals, preventing the masks of the LUTs from being combined, wherein N is an integer.

8. (currently amended) The method of claim 7 further comprising:

before determining if the LUTs implement the same function, determining if the LUTs have more than M unique input signals; and

if the LUTs have more than M unique input signals, preventing the masks of the LUTs from being combined, wherein M is an integer.

9. (original) The method of claim 1 wherein determining if the LUTs both perform the same function further comprises:

determining if an output value of the first LUT equals an output value of the second LUT for each possible input value that is applied to the input terminals of both of the LUTs.

10. (original) The method of claim 1 further comprising:

selecting third and fourth LUTs from the design;

determining if both of the third and the fourth LUTs implement a same function;

if the third and the fourth LUTs implement the same function, combining masks of the third and the fourth LUTs into a shared mask in the design.

11. (original) The method of claim 10 further comprising:

if the third and the fourth LUTs do not implement the same function, rearranging an order of at least two input signals of the third LUT with respect to input terminals of the third LUT to provide a first rearranged order; and

determining if both the third and the fourth LUTs implement the same function based on the first rearranged order of the third LUT input signals.

12. (original) The method of claim 1 further comprising:

breaking apart the mask of the LUTs if the mask lies in a critical path in the design and placing the first and the second LUTs into different logic elements within the design.

13. (currently amended) The method of claim 1 further comprising:

after determining whether both of the LUTs implement the same function, determining if LUTs have at least N common input signals; and

determining whether the LUTs have no more than M unique input signals, wherein the LUT masks are combined into the shared LUT mask only if the LUTs have at least N common input signals and no more than M unique input signals, wherein N and M are integers.

14. (original) A computer system that combines lookup tables (LUTs) in a design for a programmable integrated circuit, the computer system comprising:

code for selecting first and second LUTs in the design;

code for comparing output values of the LUTs to determine if the LUTs generate an identical function; and

code for combining masks of the LUTs in the design if the LUTs generate the identical function.

15. (currently amended) The computer system of claim 14 further comprising:

code for determining if the first and the second LUTs have at least N common input signals, before determining whether the first and the second LUTs implement the identical function, wherein N is an integer; and

code for ordering input signals of at least one of the LUTs so that each of the common input signals is applied to a corresponding LUT input terminal in the same order to provide a first rearranged order of the input signals, if the LUTs have at least N common input signals,

wherein the code for comparing the output values of the LUTs determines whether the LUTs implement the identical function based on the first rearranged order of the input signals.

16. (original) The computer system of claim 15 further comprising:

code for rearranging an order of at least two uncommon input signals with respect to input terminals of one the LUTs to provide a second rearranged order, if the LUTs do not implement the same function; and

code for comparing output values of the LUTs to determine if the LUTs generate the identical function based on the second rearranged order of the input signals.

17. (original) The computer system of claim 14 further comprising:

code for rearranging an order of input signals of one of the LUTs with respect to input terminals of that LUT to provide a first rearranged order of the input signals, if the LUTs do not generate the identical function; and

code for determining if the LUTs implement the identical function based on the first rearranged order of the input signals.

18. (original) The computer system of claim 17 further comprising:

code for combining the masks of the LUTs in the design if the LUTs implement the identical function with the first rearranged order; and

code for rearranging the order of the input signals of one of the LUTs with respect to the input terminals of that LUT to provide a second rearranged order of the input signals, if the LUTs do not implement the identical function with the first rearranged order.

19. (original) The computer system of claim 18 further comprising:

code for combining the masks of the LUTs in the design if the LUTs implement the identical function with the second rearranged order; and

code for rearranging the order of the input signals of one of the LUTs with respect to the input terminals of that LUT to provide a third rearranged order, if the LUTs do not implement the identical function with the second rearranged order.

20. (currently amended) The computer system of claim 14 further comprising:

code for determining if the LUTs have at least N common input signals; and

code for determining if the LUTs have no more than M unique input signals,

wherein implementation of the code for comparing the output values of the first and second LUTs is prevented if the first and second LUTs have less than N common input signals or more than M unique input signals, wherein N and M are integers.

21. (original) The computer system of claim 14 further comprising:

code for breaking apart the mask of the LUTs if the mask lies in a critical path in the design and placing the first and the second LUTs into different logic elements within the design.

22. (currently amended) The computer system of claim 18 further comprising:

code for determining if the LUTs have at least N common input signals, after the output values of the LUTs have been compared;

code for determining if the LUTs have no more than M unique input signals, after the output values of the LUTs have been compared,

wherein the masks of the first and the second LUTs are not combined if the LUTs less than N common input signals or more than M unique input signals, wherein N and M are integers.